Attorney's Docket No: 019322-340

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Ingo Molnar	)	
Serial No.:	09/934,738	)	
Filed:	August 22, 2001	)	**Confirmation No.** 9016
Group Art Unit:	2145	)	
Examiner:	Aziziul Q. Choudhury	)	
Attorney Docket:	019322-000340	)	
Title:	EMBEDDED PROTOCOL OBJECTS	)	

Box AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sirs/Madam:

In response to the Final Office Action dated November 24, 2009, a Notice of Appeal is being filed with the present Appeal Brief.

With regard to payment of Appeal Brief and Notice of Appeal fees, Appellant believes no fees are due at this time. Appellant previously submitted an Appeal Brief on March 12, 2008 along with a Notice of Appeal and paid the requisite fees therefor. On June 23, 2008, the Examiner promptly issued a new Office Action and thus, reopened prosecution prior to the Board of Appeals ("Board") rendering a decision on the merits. Thus, Appellant previously paid for a decision by the Board on the merits of an appeal but never received the same. Accordingly, it is submitted that Appellant need not repay the Appeal Brief and Notice of Appeal fees previously paid on March 12, 2008.

Based on the above, it is not believed that any additional fees are required. However, any fees required to allow consideration of this paper are hereby authorized to be charged to Deposit Account No. 13-4365. Additionally, it is not believed that any extensions of time are required. However, in the event that extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 CFR § 1.136(a).

Real Party in Interest

Red Hat, Inc. is the real party in interest.

Related Appeals and Interferences

There are no other appeals or interferences, known to the Appellant, or Appellant's legal representatives, which will directly affect or be directly affected by or have a bearing on the Board's decision in this pending appeal.

Status of Claims

Claims 1-14 are pending. The November 24, 2009 final rejection of all pending claims is being appealed herein.

Status of Amendments

There were no amendments filed after the Final Office Action of November 24, 2009. Applicant chose to proceed directly with this appeal. All previous papers filed by Applicant has been entered.

Summary of Claimed Subject Matter

The claimed invention improves the performance of data-serving applications by providing a mechanism to respond to a communication request made by a client application via a dynamic response. The dynamic response is created by mixing together dynamic protocol objects (i.e. the portion of the response that changes) and static protocol objects (i.e. the part of the response that is static) at the server. In effect, the part of the response that is static can be embedded in the dynamic response, along with the portion of the response that changes. If a response, such as a Web page, requires updated content, only a small portion of which has changed, most of the response can be created and sent using the cached static parts, resulting in the response being sent more quickly and efficiently than is possible with prior-art server systems.

The invention allows a communication server to respond to an application protocol request received from a client application so that the response includes at least a portion that changes.

The response is designed to be able to also include embedded static parts. In responding to the application protocol request, the portion(s) of the response that changes is sent to the client application. To complete the response, the part(s) of the response that is static is retrieved from cache and sent to the client application. The part(s) of the response that is static is effectively embedded in the dynamic response.

Claims 1, 5, 9, and 11 are commensurate independent claims that stand rejected under the same art. Claim 1 is an independent method claim. The first element of claim 1 is directed to receiving from a client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a portion of the response that is static. This element is shown as 301 in Fig. 3 and discussed in the originally-filed specification in paragraph [0027], page 12, lines 3-10. The second element of claim 1 is directed to creating at the server the portion of the response that changes. This element is shown as 303 in Fig. 3 and discussed in the specification in paragraph [0028], page 12, lines 19-22. The third element of claim 1 is directed to sending the portion of the response that changes to the client application. This element is shown as 304 in Fig. 3 and is also discussed in the specification in paragraph [0028], page 12, lines 22-23. The fourth element of claim 1 is directed to retrieving the portion of the response that is static from a cache disposed in an operating system kernel. This element is shown as 305 in Fig. 3 and 105 of Fig. 1; also, this element is discussed in the specification in paragraph [0028], page 12, lines 23-25. The fifth element of claim 1 is directed to sending the part of the response that is static to the client application. This element is shown as 306 in Fig. 3 and discussed in the specification in paragraph [0028], page 12, line 25 through page 13, line 2.

Claim 5 is a computer program product claim. It is commensurate with claim 1, discussed immediately above, and thus support and discussion of the operation of the computer program product as claimed occurs in the specification and drawings as discussed above. In addition, portions of the instructions claimed can reside in the system shown in Fig. 6 in processor 602, memory 605, and various media, such as a fixed disk drive 607, a diskette drive 608 and a display 609. This system and the nature of the computer program product, in general, is further discussed in the specification at paragraph [0040], which corresponds to page 18, line 12 through page 19, line 8. The instructions are also shown in the Appendix.

Claim 9 is an apparatus claim directed to responding to a client application. The first element of claim 9 is a cache disposed in an operating system kernel. Support for this element occurs in 105 of Fig. 1 and in paragraph [0022] of the specification (page 7, line 23 through page 8, line 5). The remainder of the elements of claim 9 are recited in means-plus-function language pursuant to 35 U.S.C. § 112 ¶ 6. In each case, the means for performing the function is a combination of a computer system like that shown in Fig. 6 and discussed in paragraphs [0020] and [0025] of the specification and a computer program product as discussed in paragraph [0019] of the specification. The function performed on each element of claim 9 is commensurate to one of the elements of claim 1 and support and discussion for each function can be similarly found.

Claim 11 recites an instruction execution system that is operable as a communication protocol server. The instruction execution system is supported by paragraph [0041] and Fig. 6. The elements of claim 11 are commensurate with those of claim 1, discussed above, and thus support and discussion of the operation of the instruction execution system occurs in the specification and drawings as discussed above.

Grounds of Rejection to be Reviewed on Appeal

Claims 1-14 are rejected as obvious under 35 U.S.C. § 103(a) by U.S. Patent 6,256,712 to Challenger et al. ("Challenger") in view of U.S. Patent 6,934,761 to Curtis et al. ("Curtis").

Arguments

I. Claims 1-14 Are Patentable Over Challenger and Curtis.

In order to establish obviousness, the Examiner must consider all of limitations of a claim in determining patentability of that claim against the prior art. M.P.E.P. § 2143.03. All of Applicant's claims, either directly, or through dependency, have recitations that cannot be found in Challenger or Curtis.

Among others, the following recitations can be found in all of claims 1, 5, 9 and 11 directly, and in all of claims 2-4, 6-8, 10 and 12-14 through dependence: 1) having a server respond to an application protocol request by sending a portion of the response that changes to the client application; and 2) retrieving a part of the response that is static from cache in an operating system kernel and *then* sending the part that is static to the client application. Nowhere

does Challenger or Curtis discuss these claim features. Challenger only discusses "maintaining updated caches and making consistent updates" to these caches, which are located on a server. *See* Challenger, column 2, lines 53-55. Curtis only discusses a method for managing HTTP caches and transmitting HTTP request data in a web server. *See* Curtis, Abstract.

All of Applicant's claims recite "receiving from the client application" an "application protocol request" and a "response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static." Challenger, by contrast, does not even mention any application protocol request, let alone discuss responding to application protocol requests. Challenger only discusses combining objects into larger objects as an update mechanism for caches. There is simply no need for an application protocol request from a client in Challenger because Challenger teaches that cached objects on a server are "immediately replaced with fresh objects" when they become stale "rather than deleting stale items from the cache." (See Challenger col. 2, line 66 - col. 3, line 1). Additionally, Challenger does not mention static or dynamic portions of responses or even Web pages, only completely static or completely dynamic Web pages.

All of Applicant's claims recite the retrieval of the part of the response that is static from a cache disposed in an operating system kernel. Applicant is at a loss to find this concept disclosed in Challenger. The portions of Challenger cited by the Examiner discuss either a proxy cache or a processor cache, neither one of which resides in a kernel. A proxy cache resides in user space and a processor cache resides inside the processor hardware. Challenger does not even mention a kernel, let alone an in-kernel cache. The Examiner has suggested that a kernel was "inherent" in Challenger because Challenger mentioned a "computer" and all current computers use an operating system and all current operating systems require a kernel. However, even if this was true, Challenger still does not mention the in-kernel cache or the very specific concept of retrieving a part of a response that is static from a cache disposed in an operating system kernel. For a proper rejection, it is not enough for the Examiner to simply analogize specific claim recitations with portions of the cited reference in a conclusory fashion. The Examiner has suggested that some unwritten "spirit of the design" behind a cited reference can be used to reject a claim over the cited reference under Section 103. However, for a proper rejection under Section 103, the Examiner must consider all of the claim limitations in

determining patentability of that claim against the prior art. M.P.E.P. § 2143.03. Applicant respectfully submits that the Examiner has failed to meet this burden.

It is noted that the November 24, 2009 Final Office Action has attempted to cite Curtis as teaching an in-kernel cache. While Curtis does mention the term "in-kernel HTTP cache," there is no disclosure in Curtis about retrieving static portions of a response from an in-kernel cache, as is specifically recited in the claims. The mere mentioning of a term in a claim by a cited reference does not disclose the complete claim recitation as arranged and taught in the claims. In this regard, it is respectfully submitted that neither Challenger nor Curtis, either singly or in combination, teach the retrieval of the part of the response that is static from a cache disposed in an operating system kernel.

Further, Challenger does not discuss "sending the portion of the response that changes to the client application and *then* retrieving the part of the response that is *static*. . . ." Challenger is only interested in constantly updating data content that has *changed* and validating webpages on the server. Challenger does not discuss sending a portion of the response that changes and *thereafter* retrieving the portion of the response that the static. Challenger only discusses validating whole webpages. Challenger does not discuss sending portions of a response at separate times. It appears as though the Examiner has ignored these claim recitations directed to time-ordering of the response, which is certainly not taught by any of the cited art. Neither the cited portions in the Final Office Action nor the "Response to Arguments" section of the Final Office Action address the above-discussed time-ordering of the response. For at least this reason alone, the Examiner had not made a *prima facie* case of obviousness.

In addition, Challenger only discloses invalidating cached objects in databases. For example, Challenger discusses invalidating cached objects having dependencies on records which have changed, where the records are portions of the database. However, nowhere does Challenger teach or suggest sending a response by sending a portion of the response that changes and then retrieving and sending a portion of the response that is static, as recited in the claims. As previously admitted by the Examiner, Challenger does not even disclose a response. However, the Examiner suggests the webpages in Challenger and Curtis can be interpreted as such response. Nonetheless, Challenger does not teach or disclose creating a webpage with a portion that changes and a portion that is static. Challenger only discloses completely dynamic or

completely static webpages. In light of the above, Challenger cannot render Applicant's claims obvious. Curtis does not remedy the deficiencies of Challenger. Indeed, the webpages discussed in Curtis are all sent at one time and all webpages are also either completely static or completely dynamic webpages. There is no discussion in Curtis of sending a webpage having a partially dynamic portion as well as a partially static portion. Further, there is certainly no discussion in Curtis of sending a response to a request by sending the dynamic portion of the webpage and then retrieving and sending the static portion of the webpage.

For at least the above reasons, Applicant submits that every element of any claim is not taught or suggested by the cited art. Applicant believes he has responded to the concerns raised by the Examiner. Reconsideration of this application, as amended, is hereby requested.

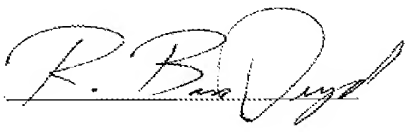
The recitations already discussed in independent claims 1, 5, 9, and 11 are also contained in dependent claims 2-4, 6-8, 10 and 12-14 through their dependence therefrom. Thus, claims 2-4, 6-8, 10 and 12-14 are also patentable for at least the reasons presented above.

* * * * *

Conclusion

For the reasons state above, Applicant respectfully submits that the rejection standing in this application is improper. The Examiner has failed to establish a prima facie case of obviousness under 35 U.S.C. § 103(a). Therefore, Applicant respectfully submits that claims 1-14 are in condition for allowance. Accordingly, the reversal of the rejection of claims 1-14 is respectfully requested.

Respectfully submitted,

Date: 2/24/2010 By: 

R. Brian Drozd
Registration No. 55,130,
Attorney for Applicant

Moore & Van Allen, PLLC
P.O. Box 1370A6
Research Triangle Park, NC 27709
Telephone: (919) 286-8140
Facsimile: (919) 286-8199

Claims Appendix

The following is a clean copy of the claims involved in this appeal.

1. In a communication server, a method of responding to a client application, the method comprising the steps of:

receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static;

creating at the server the portion of the response that changes;

sending the portion of the response that changes to the client application and then retrieving the part of the response that is static from a cache disposed in an operating system kernel; and

sending the part of the response that is static to the client application.

2. The method of claim 1 wherein the cache disposed within the operating system kernel is a protocol object cache.

3. The method of claim 1 wherein the application protocol request and the reply are formatted according to a hypertext transfer protocol (HTTP).

4. The method of claim 2 wherein the application protocol request and the reply are formatted according to a hypertext transfer protocol (HTTP).

5. A computer program product comprising at least one of a CD-ROM, DVD-ROM, magnetic tape, diskette, magnetic fixed disk and a semiconductor device having computer program code embodied therein, the computer program code for enabling a server to respond to a client application, the computer program code comprising:

instructions for receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static;

instructions for creating at the server the portion of the response that changes;

instructions for sending the portion of the response that changes to the client application and then retrieving the part of the response that is static from a cache disposed in an operating system kernel; and

instructions for sending the part of the response that is static to the client application.

6. The computer program product of claim 5 wherein the cache disposed within the operating system kernel can be a protocol object cache.

7. The computer program product of claim 5 operable to format the application protocol request and the reply according to a hypertext transfer protocol (HTTP).

8. The computer program product of claim 6 operable to format the application protocol request and the reply according to a hypertext transfer protocol (HTTP).

9. Apparatus for responding to a client application, the apparatus comprising:

a cache disposed in an operating system kernel;

means for receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static;

means for creating at the server the portion of the response that changes;

means for sending the portion of the response that changes to the client application and then retrieving the part of the response that is static from the cache through an operable connection to the cache; and

means for sending the part of the response that is static to the client application.

10. The apparatus of claim 9 wherein the cache can be a protocol object cache.

11. An instruction execution system operable as a communication protocol server, operable to respond to a client application by performing the steps of:

receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static;

creating at the server the portion of the response that changes;

sending the portion of the response that changes to the client application and then retrieving the part of the response that is static from a cache disposed in an operating system kernel; and

sending the part of the response that is static to the client application.

12. The instruction execution system of claim 11 further operable as a hypertext transfer protocol (HTTP) server.

13. The instruction execution system of claim 11 wherein the cache can be a protocol object cache.

14. The instruction execution system of claim 12 wherein the cache can be a protocol object cache.

Evidence Appendix

None

Related Proceedings Appendix

None